

# GarmNet: Improving Global with Local Perception for Robotic Laundry Folding

Daniel Fernandes Gomes<sup>1</sup>[0000–0001–8619–4966], Shan Luo<sup>1</sup>[0000–0003–4760–0372],  
and Luis F. Teixeira<sup>2,3</sup>[0000–0002–4050–7880]

<sup>1</sup> Department of Computer Science,  
University of Liverpool, Liverpool, UK  
{danfergo,shan.luo}@liverpool.ac.uk

<sup>2</sup> Faculdade de Engenharia, Universidade do Porto, Portugal

<sup>3</sup> INESC TEC, Porto, Portugal  
luisft@fe.up.pt

**Abstract.** Developing autonomous assistants to help with domestic tasks is a vital topic in robotics research. Among these tasks, garment folding is one of them that is still far from being achieved mainly due to the large number of possible configurations that a crumpled piece of clothing may exhibit. Research has been done on either estimating the pose of the garment as a whole or detecting the landmarks for grasping separately. However, such works constrain the capability of the robots to perceive the states of the garment by limiting the representations for one single task. In this paper, we propose a novel end-to-end deep learning model named *GarmNet* that is able to simultaneously localize the garment and detect landmarks for grasping. The localization of the garment represents the global information for recognising the category of the garment, whereas the detection of landmarks can facilitate subsequent grasping actions. We train and evaluate our proposed GarmNet model using the CloPeMa Garment dataset that contains 3,330 images of different garment types in different poses. The experiments show that the inclusion of landmark detection (GarmNet-B) can largely improve the garment localization, with an error rate of 24.7% lower. Solutions as ours are important for robotics applications, as these offer scalable to many classes, memory and processing efficient solutions.

**Keywords:** Garment Localization · Landmark Detection · Robot Laundry Folding

## 1 Introduction

Garment recognition is one necessary capability not only for the automation of tasks involving its manipulation, such as garment folding, within robotic systems [15] but many other applications as well: online e-commerce platforms [23] that make suggestions based on image information [12], intelligent surveillance systems that track people based on the clothing description, etc. However, recognition of clothes, or (highly) deformable objects in general, is a challenging task due to the many poses and deformations that a flexible object may exhibit.

We consider the scenario wherein an image containing a single piece of clothing, flat, wrinkled and semi-folded exists on a clean background and a robotic system wants to find the garment and good grasping points. Our goal is then to perceive, at the global level, a piece of garment existing in an image, by localizing and classifying it. And, at a local level, by identifying and localizing its landmarks, e.g., neckline-left, right-armpit, right-sleeve-inner, is an image point. Each garment class has a different amount and types of landmarks e.g., towels have four, whereas t-shirts and long t-shirts have both 12. Because of such variances, garment+landmark detection can be formulated using two different approaches: (a) garment finding as object localization, followed by conditional, class specific, landmarks finding, also as object localization. (b) Finding all landmarks existing in the image independently of the garment class as object detection, and the garment piece as object location.

Although approach (a), is a simpler solution to build using off-the-shelf models, and is commonly seen in current literature [11] [16], it is more inefficient because: 1. It requires one different sub-model for each garment category, being that many landmarks are shared between garment category e.g., a sleeve of a hoody is similar to a sleeve in a jacket. 2. On the other hand, when using Neural Networks, these work by building more complex representations as its depth increases, through the combinatorial effect of chaining multiple layers together. Therefore, a Network that recognizes a piece of clothing, should in principle, recognize somewhere in its hidden layers some of its landmarks. This means that in approach (a) multiple redundant hidden features are learned, representing extra parameters to be stored in memory, and more operations to be computed during execution time. Adding to this, in the context of robotics e.g., a top view of a laundry bin, the garment global perception might not be possible with good accuracy, but recognizing some local landmarks might be just as valuable for the robot, as it could grasp the piece of clothing by one good landmark and then perform further recognition using the same model.

Therefore, we address the detection of landmarks and classification+localization of garment simultaneously, with a Convolutional Neural Network (CNN) [9] composed of one common trunk and two separate branches. We then introduce a bridge connection that feeds the landmarks detection output into the garment localizer branch, resulting in a decrease from 56.7% to 32.0% in the error rate, that demonstrates the advantages in considering both tasks together. We balance and augment our dataset with Gaussian and hue noise, and perform one last training achieving: 0% and 17.8% error rate on classification and classification+localization respectively; and 36.2% mean Average Precision (mAP) on landmark detection.

## 2 Related Work

*Early works* handling clothes happen in Robotics with the folding task, and the problem domain is constrained enough to avoid the necessity of classification i.e., only one type of clothing is considered. In [15] towels are considered, and

depth discontinuities are explored to detect its borders and corners. With that information, a PR2 robot is able pick them from a random dropped position and, following a predefined sequence of steps, folds them.

*Machine Learning methods* are latter used, not only in robotic tasks but other software applications as well e.g., in [23] real-time classification and segmentation of people’s clothing appearing in a camera video feed are addressed. However, using the raw image i.e., each pixel is a feature, would result in a low performance results due to the curse of dimensionality, that many Machine Learning (ML) methods suffer from. To overcome this challenge, one common approach is to use the Bag of Visual Words (BoW), that extracts handcrafted features e.g., Scale-Invariant Feature Transform (SIFT) or Histogram of Oriented Gradients (HOG) and feeds these into a classifier e.g., Support Vector Machine (SVM) or k-Nearest Neighbours (k-NN). In [11] the authors use this approach to design a two layer hierarchical classifier to determine the category and grasping point (equivalent to pose, in this work) of a piece of clothing. Other works address the extraction of domain specific features from images. In [22], a set of Gabor filters are used to filter edges magnitude and orientation that are representative of wrinkling and overlaps and with that information, the authors propose three types of features: Position and orientation distribution density, cloth fabric and wrinkle density; and existence of cloth-overlaps.

*Deep Learning (DL) methods.* In 2012, AlexNet [8] achieves a notorious improvement of 11% in the ILSVRC2012 image classification competition, when comparing against the next best solution. In [6], to address object detection, region proposal techniques are combined with CNNs, resulting in R-CNN, an hybrid method that combines a CNN with SVMs. Then, in [5], two main improvements are made: the RoI pooling layer is introduced, and the SVM is replaced by a softmax classifier. The improved model Fast R-CNN model is then a two headed CNN, optimized using a multi-task loss,  $9\times$  quicker to train and  $213\times$  faster test-time. In [19], the authors further introduce Region Proposal Networks (RPNs), making this architecture finally an end-to-end trainable model. Another simpler architecture, YOLO, is introduced in [17] and improved in [18], consisting of only two direct branches on the top of a regular fully CNN: one for positions coordinates regression and another for classes attribution. The system is capable of real-time object detection.

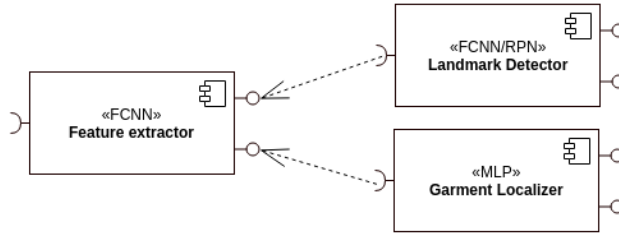
*DL in garment perception.* After the successes using DL methods, some works that address garment perception also explore their potential, mainly considering classification problems. One example is [16], that addresses the same problem as [11]: pose and category recognition. The solution is also similar, a two layer hierarchical classifier. But here, instead of BoW and SVMs, one CNN is used to determine the garment class, followed by a class specific CNN that determines the garment pose index. The authors compare the model against others using hand engineered features and report gains of 3% in accuracy. Similarly, in [1] a robotic system composed of two arms and a RGB-D camera, uses an hierarchy

of CNNs with three specialization levels. At a first step one CNN classifies the garment piece, and then two others are used to find the first and second grasping points.

*Our approach.* In contrast to these approaches, ours leverages the intrinsic characteristics of CNNs and the architecture patterns from both classification/localization and detection models to perform the global and local perception in one step with a single CNN. Our model can also exploit prior knowledge gained to detect landmarks, to enhance the global garment perception. Our approach is therefore more memory and processing efficient than hierarchical solutions presented above. This happens because, in our approach, lower level layers are shared between the Global and Local perception components, as discussed in section 1. Exploring these intrinsic characteristics of CNNs, in [20], the authors propose Hierarchical Convolutional Neural Networks (H-CNN), to address hierarchical classification. With coarse/top-level classifications being extracted from hidden layers while finer/more-specific classes being predicted by the latest layers of the network.

### 3 Network architecture

Our network, **GarmNet**, at a macro level can be summarized into three blocks: **Feature extractor**, **Landmark detector** and **Garment localizer**.



**Fig. 1. GarmNet macro view**, UML [3] Components Diagram. The architecture is broken into three blocks (components): Feature extractor, Landmark Detector and Garment Localizer; that output: intermediate features at two depths, landmarks classes+localizations and garment class+localization.

*Feature extractor* We implement the feature extraction module with a Fully Convolutional Neural Network (FCNN), a 50-layer ResNet[7]. The model is pre trained on ImageNet[2], to which we remove the last Fully Connected (FC) layers, resulting in a  $7 \times 7$  output tensor. Yet, because in some cases we have multiple landmarks close to each other, we preferred a larger output size, that

would result in a higher number of anchors in the landmark detector. We achieve this by probing the ResNet at the end of the *conv4\_x* block, which has an output size of  $14 \times 14$ .

*Landmark detector* Responsible for classifying and localizing all the landmarks present in the image, this module is a small,  $3 \times 3$ , sliding FCNN, similar to the Region Proposal Network (RPN) introduced in [19], and it is implemented with convolutional (CONV) layers. The network is composed by an intermediate 256-d layer, with a rectified linear unit (ReLU) [8] activation, followed by two heads: one for localization and other for classification, see 2.

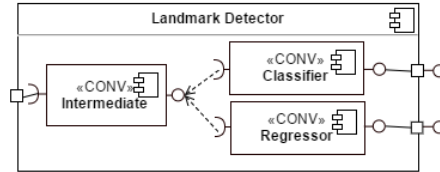
The localizer head is a Multi-Layer Perceptron (MLP), that outputs the predicted landmark relative coordinates such that

$$A_{(i,j)} = (s * j, s * i) \quad (1)$$

$$L_{(i,j)} = A_{(i,j)} + O_{(i,j)} \quad (2)$$

where  $L$  is a predicted landmark location,  $A$  stands for the sliding network position  $O$  the localization head output, and  $s$  a stride value that we define to spread the base referential (or anchor points, as introduced in [19]) evenly across the input image (we set it as 18, that together with a 26 landmark area matches the 224 image input size).

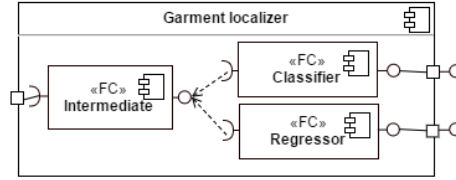
The classifier head, a convolutional layer with  $n + 1$  filters, where  $n$  is the number of landmark classes and the plus one answers for background, non positive, landmarks. We apply a *softmax* activation along the depth dimension and, therefore the output of this layer can be interpreted, at each position, as the probability of the associated landmark  $L_{(i,j)}$  being of a certain class, or background.



**Fig. 2. Landmark detector** component, UML[3] representation. After one intermediate branch, two separate branches output  $18 \times 18$  landmark proposals (classification and location). This block, implemented with convolutional layers, can be interpreted as small fully connected network sliding over the feature extractor output.

*Garment localizer* To perform the localization of the piece of clothing present in the image, we use a two head three-layer fully connected network, similar to the sliding window used for landmark detection. The intermediate layer is a 512-d FC

with ReLU activation, followed by the regression and classification heads. The regression head outputs four values:  $x$ ,  $y$ , width and height; and the classification head outputs  $n$  values, where  $n$  is the number of garment classes, that we remap to probabilities with a *softmax* activation. We retrieve the predicted landmarks by computing the *argmax* over the classifier output tensor depth dimension, and associating it with the point predicted in same spacial position on the regression head. We further discard all the landmarks that have a confidence value i.e., the value that motivated the argument to be the maximum, lower than 0.5.



**Fig. 3. Garment localizer component**, UML[3] representation. Similar to the landmark detector 2, yet fully connected layers are used. The Intermediate layer outputs a 512-d, the classifier a 9-d (one hot encoded classes) and the regressor the 3-d ( $x$ ,  $y$ , width and height) vectors.

## 4 Experiments

Our implementation was performed using Keras<sup>4</sup> framework with the TensorFlow<sup>5</sup> back-end. All experiments were carried out on a laptop, with GPU support enabled, equipped with an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 16GB DDR4 RAM, and a Nvidia Geforce GTX 1060 6GB GDDR5 GPU. We initialize kernels with random values drawn from a normal distribution, and bias with ones. Optimization is performed using Adadelta with 1.0 learning, a batch size of 30, and 40 epochs per experiment. At test time our model runs at roughly 30 FPS. For classification and localization evaluation we use error rate, while for detection we use the mean Average Precision metric (mAP) as proposed in [4]. The source code has been made publicly available<sup>6</sup>.

### 4.1 Dataset

We adapt the CTU Color and Depth Image Dataset of Spread Garments [21]. This dataset is divided into two groups: “Flat and wrinkled”, with 2050, and

<sup>4</sup> <https://keras.io/>

<sup>5</sup> <https://www.tensorflow.org/>

<sup>6</sup> <https://github.com/danfergo/garment>

“Folded”, with 1280 examples. Each example contains one image of a piece of clothing placed on a wooden floor and is annotated with the stereo disparity; garment category; interest points coordinates and category; and other meta-information. We merge both groups, and because it only contains information regarding each landmark position, we extend its annotation with the garment bounding box as follows:

$$P_1 = (\min_{l \in L} l_i, \min_{l \in L} l_j) \quad (3)$$

$$P_2 = (\max_{l \in L} l_i, \max_{l \in L} l_j) \quad (4)$$

where  $L$  is the set of Landmarks,  $P_1$  and  $P_2$  are the top-left and bottom-right corners of the bounding box. There is a total of 27 landmark categories, distributed among 9 types of garment. Some landmark categories are shared among classes. We then create two splits, with 300 randomly chosen images for validation and the remaining the ones used for training. The remaining 2318, make the training split. Results are reported over the validation split.

## 4.2 Landmark detection anchors

For training the landmark detector heads we transform the landmark locations into small squared areas and follow a strategy similar to the anchor boxes described in [19]. To all the anchor boxes that intercept a landmark box with  $IoU > 0.7$ , we consider it a positive for the respective class. If it does not intercept any with  $IoU \geq 0.3$  we set it to background. Because the ratio between positive and negative anchor boxes is high, we create a binary mask that is used to filter the anchors that effectively are considered in the loss function. This mask selects all the class positive and 10 randomly chosen background anchors.

## 4.3 Loss functions

*Landmark detector* For the localization head, we apply the robust loss defined in [19] to all the active anchors that are selectable by the mask,  $m$ , described in 4.2.

$$L_{reg} = m \odot L_{robust}(P, T) \quad (5)$$

$$L_{robust}(P, T) = \begin{cases} (P - T)^2, & P - T < 0.5 \\ |P - T|, & \text{otherwise} \end{cases} \quad (6)$$

For the classification head, we apply the cross-entropy loss function to all active anchors, being that at each anchor the landmark class is one-hot encoded.

$$L_{cls} = m \odot L_{ce}(T, P) \quad (7)$$

*Garment localizer* With the garment classes represented in a one hot encoded vector, we use the cross-entropy loss on the classification head and, for the regression head, we use the mean squared error.

#### 4.4 One landmark class per sample, constraint

One important peculiarity of landmark detection to consider is the fact that, per image, only one class of landmarks exist. Therefore, we can introduce this constraint into the loss function and promote parameter combinations that tend to predict only one landmark per class. We implement this constraint by also applying cross-entropy over the spacial dimension, resulting in the loss function 8. However, because cross-entropy expects its input to be a probability distribution, we must firstly apply *softmax* accordingly. We therefore, place two *softmax* after the last convolutional layer activation: the first, (regular) depth wise, the second, spacial wise; and pass each output to the correspondent cross-entropy. At test time, we average the two *softmax*, similarly to 8. Because for each garment category, only a few landmarks are active, we further create a second mask, that is the ground-truth max spacial value, and we use it to ignore the loss spacial component for the landmark classes that are not applicable.

$$L_{cls} = \frac{1}{2} \left[ m \odot L_{ce}(T, P) + \max_{w \times h} T \odot L_{ce(spatially)}(T, P) \right] \quad (8)$$

Although with the spacial constraint loss addition, we achieve a 2% lower detection mAP score, dropping from 37.8% to 35.7%. Yet, we are able to achieve lower duplicated predictions, as illustrated in 4.

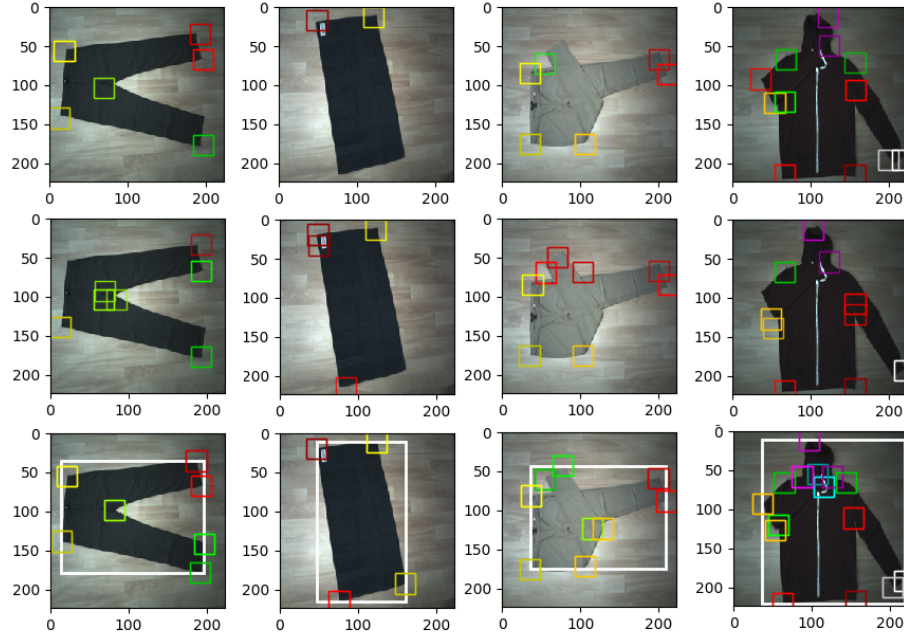
#### 4.5 Using landmarks within garment localization

We investigate the gains in feeding the landmark detector output features into the garment localize intermediate layer, expecting that these would help better frame the garment bounding box and. We flatten the  $18 \times 18$  tensor outputted by the classifier block from the landmark detector branch, and concatenate it with the flatten Feature Extractor output, before feeding it to the 512-d intermediate layer, resulting in GarmNet-B, represented in the figure 5. With this bridge connection, the network achieves 32.0% classification+localization error rate, a 24.7% improvement when comparing with the individual garment detector training.

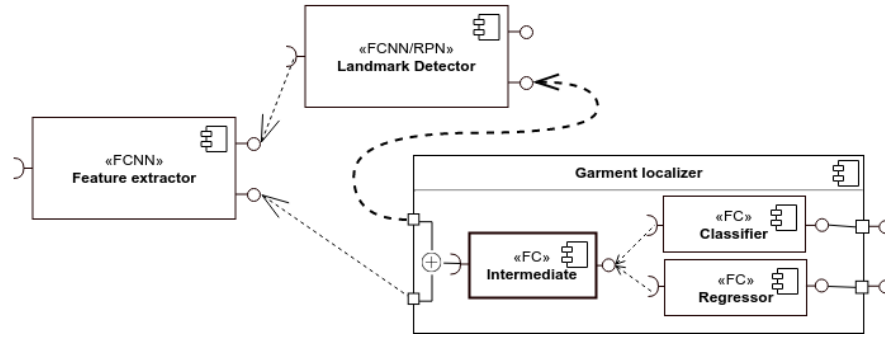
**Table 1.** Summary of Classification and Classification+Localization error rates. GarmNet is our base model, GarmNet-B is the model obtained after the bridge connection introduced in 4.5 and GarmNet-B (A.D) the latter model optimized using the augmented dataset.

	Classification	Classif.+Loca.
GarmNet	0%	56.7%
GarmNet-B	0%	32.0 %
GarmNet-B (A.D.)	0%	17.8%





**Fig. 4.** Representative cases of the result of applying the spacial constraint loss. At the top row, predictions with composed loss, at the middle, without, and the bottom, the ground truth.



**Fig. 5.** GarmNet-B (introduced in 4.5) representation using UML[3]. The output emitted by the classifier block from the Landmark Detector branch is concatenated with the Feature Extractor output before being fed into the intermediate layer.

#### 4.6 Final optimization with Augmented Data

We perform one last training, without loading any previous learned parameters (with the exception of the feature extractor ImageNet parameters) and using augmented data. The data augmentation is achieved by repeating examples of less numerous classes and adding Gaussian and hue noise. The obtained results are: 0% classification and 17.8% classification+localization error rate, and 36.2% landmark detection mAP. The complete classification accuracy can be justified by the almost constant background and the few, often differently colored, garments per class.

### 5 Conclusion

In this work, we proposed a novel deep neural network model named GarmNet that can be optimized in an end-to-end manner, for performing simultaneous garment local and global perception. Approaches as ours are important for robotics applications, as these offer scalable to many classes, memory and processing efficient solutions, enabling real-time perception capabilities. We evaluate our solution using an augmented dataset assembled using the two collected by CTU during the CloPeMa project [21]. The experiments showed the effectiveness and side effects of introducing domain specific knowledge into the loss function being optimized, at both quantitative and qualitative levels. We finally demonstrate the improvements on garment localization by considering the landmark detection as an intermediate step.

In the future work, more experiments will be done to further optimise the network architecture and its hyper-parameters configuration. A more challenging dataset, with higher number of images and variability e.g., [12], will also be used. Within the context of garment perception for robotic laundry folding, the work will be extended to garment folding with a robot arm-hand setup, supported by the garment perception done in this paper and also possibly assisted by tactile sensors [13, 14, 10].

### ACKNOWLEDGMENT

This work was supported by the EPSRC project “Robotics and Artificial Intelligence for Nuclear (RAIN)” (EP/R026084/1).

### References

1. Corona, E., Aleny, G., Gabas, A., Torras, C.: Active garment recognition and target grasping point detection using deep learning. *Pattern Recognition* **74**, 629 – 641 (2018). <https://doi.org/https://doi.org/10.1016/j.patcog.2017.09.042>, <http://www.sciencedirect.com/science/article/pii/S0031320317303941>
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09* (2009)

3. Engels, G., Heckel, R., Sauer, S.: Uml - a universal modeling language? LNCS (10 2000). [https://doi.org/10.1007/3-540-44988-4\\_3](https://doi.org/10.1007/3-540-44988-4_3)
4. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vision* **88**(2), 303–338 (Jun 2010). <https://doi.org/10.1007/s11263-009-0275-4>, <http://dx.doi.org/10.1007/s11263-009-0275-4>
5. Girshick, R.B.: Fast R-CNN. *CoRR* **abs/1504.08083** (2015), <http://arxiv.org/abs/1504.08083>
6. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* **abs/1311.2524** (2013), <http://arxiv.org/abs/1311.2524>
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *CoRR* **abs/1512.03385** (2015), <http://arxiv.org/abs/1512.03385>
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems* (2012)
9. Lecun, Y., Bengio, Y.: Convolutional networks for images, speech, and time-series. *The Handbook of Brain Theory and Neural Networks* (01 1995)
10. Lee, J.T., Bollegala, D., Luo, S.: "Touching to See" and "Seeing to Feel": Robotic Cross-modal Sensory Data Generation for Visual-Tactile Perception. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2019)
11. Li, Y., Chen, C.F., Allen, P.K.: Recognition of deformable object category and pose. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2014)
12. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016)
13. Luo, S., Bimbo, J., Dahiya, R., Liu, H.: Robotic tactile perception of object properties: A review. *Mechatronics* **48**, 54–67 (2017)
14. Luo, S., Mou, W., Althoefer, K., Liu, H.: iCLAP: Shape recognition by combining proprioception and touch sensing. *Autonomous Robots* pp. 1–12 (2018)
15. Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., Abbeel, P.: Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In: *2010 IEEE International Conference on Robotics and Automation*. pp. 2308–2315 (May 2010). <https://doi.org/10.1109/ROBOT.2010.5509439>
16. Mariolis, I., Peleka, G., Kargakos, A., Malassiotis, S.: Pose and category recognition of highly deformable objects using deep learning. In: *2015 International Conference on Advanced Robotics (ICAR)*. pp. 655–662. IEEE (jul 2015). <https://doi.org/10.1109/ICAR.2015.7251526>, <http://ieeexplore.ieee.org/document/7251526/>
17. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. *CoRR* **abs/1506.02640** (2015), <http://arxiv.org/abs/1506.02640>
18. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. *CoRR* **abs/1612.08242** (2016), <http://arxiv.org/abs/1612.08242>
19. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR* **abs/1506.01497** (2015), <http://arxiv.org/abs/1506.01497>

20. Seo, Y., shik Shin, K.: Hierarchical convolutional neural networks for fashion image classification. *Expert Systems with Applications* **116**, 328 – 339 (2019). <https://doi.org/https://doi.org/10.1016/j.eswa.2018.09.022>, <http://www.sciencedirect.com/science/article/pii/S0957417418305992>
21. Wagner, L., K.D., Smutn, V.: Ctu color and depth image dataset of spread garments. Tech. Rep. CTUCMP201325, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic (September 2013)
22. Yamazaki, K.: Instance recognition of clumped clothing using image features focusing on clothing fabrics and wrinkles. 2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015 pp. 1102–1108 (2016). <https://doi.org/10.1109/ROBIO.2015.7418919>, <http://dx.doi.org/10.1007/s10514-016-9559-z>
23. Yang, M., Yu, K.: Real-time clothing recognition in surveillance videos. In: Macq, B., Schelkens, P. (eds.) ICIP. pp. 2937–2940. IEEE (2011), <http://dblp.uni-trier.de/db/conf/icip/icip2011.html#YangY11>

## 6 Appendix

**Table 2.** Summary of landmark Classification+Localization, as follows: GarmNet, the results obtained using the base model; GarmNet (S.C), the base model optimized with the spacial constraint (4.4); GarmNet-B, the modified model with the bridge connection (4.5); and, GarmNet-B (A.D.), the latter model optimized using augmented dataset.

	GarmNet	GarmNet (S.C.)	GarmNet-B	GarmNet-B (A.D.)
mean AP	37.8	35.7	34.5	36.1
Left leg outer	38.5	33.7	33.0	38.2
Left leg inner	47.6	39.5	0	44.1
Crotch	46.4	42.5	43.7	34.3
Right leg inner	46.0	42.6	50.5	39.9
Left leg inner	45.3	41.4	43.7	40.9
Top right	62.1	54.7	50.5	55.7
Top left	57.3	58.4	47.8	50
Right sleeve inner	44.3	38.7	60.2	37.3
Right sleeve outer	38.2	30.4	53.6	42.2
Left sleeve inner	43.9	41.5	46.5	48.3
Left selave outer	34.5	30.7	46.0	35.7
Hood right	0	54.5	46.4	0
Hood top	0	0	42.5	40.9
Hood left	56.4	30.9	0	0
Bottom left	38.8	38.2	34.5	40.7
Bottom middle	0	0	0	43.9
Bottom right	38.6	37.1	40	39.9
Right armpit	46.1	41.7	44.7	50.7
Right shoulder	44.0	39.7	47.1	60.2
Neckline right	44.5	32.2	45.0	36.1
Collar right	44.1	38.7	47.2	48.7
Collar left	55.3	48.4	51.9	60.2
Neckline left	34.3	33.8	35.6	36.1
Left shoulder	44.4	39.1	46.7	48.7
Left armpit	43.3	38.8	39.0	37.4
Fold 1	27.5	26.4	27.5	25.2
Fold 2	0	0	0	0